

A First Book of C++

From Here to There

Class Functions, Conversions, and Inheritance

Operator Functions

- C++ operators listed in Table 14.1 can be redefined for class use, subject to certain restrictions
 - Symbols not in Table 14.1 cannot be redefined
 - `++`, `::`, and `?:` symbols cannot be redefined
 - New operator symbols cannot be created
 - Neither the precedence nor the associativity of C++'s operators can be modified

Operator Functions (continued)

- Additional restrictions on redefining C++ operators for class use
 - Operators cannot be redefined for built-in types
 - Unary operator cannot be changed to binary operator, and vice versa
 - Operator must be member of a class or be defined to take at least one class member as an operand

Operator Functions (continued)

- Defining more `Date` class operators
 - Subtraction of 2 dates is meaningful
 - Number of days between two dates
 - Comparison of two dates is meaningful
 - Does one date occur before another?
- **User-defined function:** Function that redefines C++'s built-in operator symbols for class use

Operator Functions (continued)

- Declaring and implementing operator functions: Same as for all C++ member functions except:
 - Function's name connects appropriate operator symbol to operator defined by the function
 - **Format:** `operator <symbol>`
where `symbol` is an operator listed in Table 14.1
- **Examples:**
 - `operator++` is the name of addition function
 - `operator==` is the name of comparison function

Operator Functions (continued)

- **Writing operator function:** Function accepts desired inputs and produces correct return value
- **Creating comparison function for `Date` class:** Compare two dates for equality
 - Select C++'s equality operator
 - Function name is `operator==`
 - Accepts two `Date` objects
 - Compares objects
 - Returns a boolean value indicating result
 - True for equality, False for inequality
